Tower of Hanoi extensions with emphasis on graphical representations of the problem

Saanvi Singh[1] [0009−0009−7660−6635], Sanskriti Gupta[2] [0009−0004−6401−8397], and Dhruv Jaiswal[1] [0009−0002−2629−3393]

[1] Indraprastha Institute of Information Technology Delhi (IIITD) CSE department.
[2] Indraprastha Institute of Information Technology Delhi (IIITD) CSSS department.

Abstract. This paper describes the problem called Tower of Hanoi and delves into other representations of the question with emphasis on it's relation to the Sierpiński triangle and fractals. We will discuss the significance of the constant 466/885 and the current solutions to the problem along with a few proposed paradigms which may be explored in hunt of a solution. These new paradigms include a view of how Tower of Hanoi scales up when stretched into multiple dimensions and a proposal for our method of an optimal solution of Tower of Hanoi with 4 pegs. The multiple dimensional view of tower of Hanoi comes from an extension of fractals in multiple dimensions such that solutions to the problem indicate the existence of different pathways which connect the states of the system projected onto higher dimension fractals. The analysis of the algorithmic puzzle would thus have graphical implications on the higher dimension objects and vice verse. Thus looking at the problem through a different lens may hold new understandings of problems such as higher dimension packing and Simplicial complexes in topology.

Keywords: Tower of Hanoi · Sierpiński triangle · fractals · Hanoi constant · Pascal's triangle · recursion · time complexity optimization · isomorphism · groups · induction · algorithms · cliques · graphs · projections and level sets · rotations

1 Introduction

1.1 Background and Fundamental Rules

The Tower of Hanoi puzzle, created in 1883 by French mathematician Édouard Lucas, has become a fundamental example in the study of algorithms and discrete mathematical concepts.

The origin of the Tower of Hanoi puzzle is often linked to the legend of the Tower of Brahma. According to this tale, three diamond needles hold a tower of 64 golden disks. A group of Brahmin monks is tasked with moving the entire tower to another needle, following the same rules as the Tower of Hanoi game. The legend states that the world will end when the monks complete this task.

The classic Tower of Hanoi puzzle has three wooden pegs and n disks of varying diameters. Initially, all n disks are stacked on one peg in descending order of diameter, with the largest disk at the bottom. The objective is to transfer the entire stack to a different peg, obeying the following rules:

1. Only one disk can be moved at a time.
2. Each move involves lifting the top disk from one stack and placing it onto another stack or an empty peg.
3. A disk cannot be placed on top of a smaller disk.

1.2 Mathematical Significance

The Tower of Hanoi is a classical puzzle highlighting vital mathematical concepts like recursion, algorithm design, and combinatorics. It demonstrates the power of recursive problem-solving, where each solution is built upon smaller sub problems, providing a foundation for understanding algorithmic behavior. The total number of moves required to solve the puzzle forms a geometric series: $1 + 2 + 4 + \cdots + 2^{n-1}$, which equals $2^n - 1$. This exponential growth demonstrates its connection to key mathematical concepts in computer science, such as complexity analysis.

Moreover, it showcases a divide-and-conquer strategy, a crucial approach in algorithm development. Beyond its theoretical importance, the Tower of Hanoi finds applications in practical areas such as memory allocation, sorting algorithms, and scheduling tasks. This blend of theoretical and practical utility makes the puzzle a staple in both mathematical studies and algorithm design.

2 Analysis of 3-Peg Tower of Hanoi

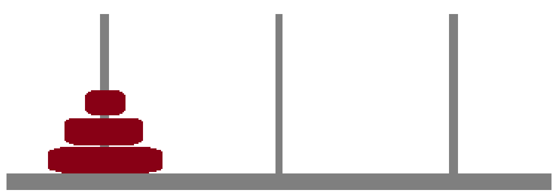

Fig. 1. Tower of Hanoi with 3 Pegs and 3 Disks

2.1 Recursive Solution for Optimal Path

Let the three pegs be labeled as *Src*, *Aux*, and *Dst*. Initially, all disks are stacked on *Src*, and the goal is to move all disks to *Dst*. For any number n of disks, the solution involves these key steps:

1. Move the top $n-1$ disks from *Src* to *Aux* (using *Dst* as an intermediary).

2. Move the bottom (largest) disk from *Src* to *Dst*.
3. Move the $n-1$ disks from *Aux* to *Dst* (using *Src* as an intermediary).

```
Solve(N, Src, Aux, Dst)

if N = 0 then
   exit
else
   Solve(N - 1, Src, Dst, Aux)
   Move from Src to Dst
   Solve(N - 1, Aux, Src, Dst)
end if
```

Recurrence Relation Let $T(n)$ represent the minimum time required to transfer n disks from *Src* to *Dst*. Based on the algorithm outlined above, the recurrence relation for this problem is:

$$T(1) = O(1) \quad \text{(base case)}$$

$$T(n) = 2T(n-1) + O(1) \quad \text{(recursive case)}$$

This relation provides an upper bound on the minimum time needed to solve the Tower of Hanoi problem with 3 pegs and n disks. However, it can be proven that this is, in fact, the optimal solution.

When moving the largest disk from *Src* to *Dst*, the remaining $n-1$ disks must temporarily be placed on *Aux*. To move these $n-1$ disks from *Src* to *Aux*, it takes at least $T(n-1)$ time. Since we are looking for the most efficient solution, we can assume this takes exactly $T(n-1)$ time. Therefore, the recurrence relation not only gives an upper bound but also represents the lower bound on the minimum time, confirming that this is the most optimal strategy.

Solving the recurrence relation yields the expression:

$$T(n) = O(2^n),$$

and the minimum number of moves required is $2^n - 1$. The result shows that the number of moves grows exponentially with the number of disks.

2.2 Representation using Hanoi Graphs

Hanoi Graphs or state graphs are undirected graphs where:

- The vertices represent all possible states of the Tower of Hanoi puzzle.
- The edges correspond to valid moves between pairs of these states.

Each vertex in the graph represents a distinct arrangement of disks on the pegs, leading to a total of k^n possible configurations. Each edge represents a valid move that transitions one configuration to another. Since the moves in the Tower of Hanoi are reversible, the corresponding Hanoi graphs are *undirected.* Additionally, as it is possible to reach any vertex from any other vertex by traversing adjacent edges, the graph is also *connected.* From this point onward, the Hanoi graph for the puzzle with n disks on k pegs will be denoted as H_n^k. A clarification in the naming convention for the vertices (1-indexed strings) in the Hanoi graph is provided as follows:

- A Hanoi graph accompanied with the prefix 'R' indicates that if the $i'th$ number of the state (from the right) is j, that would mean that in the current state of the puzzle, the $i'th$ disk is on the $j'th$ peg.
- A Hanoi graph accompanied with the prefix 'L' indicates that if the $i'th$ number of the state (from the left) is j, that would mean that in the current state of the puzzle, the $i'th$ disk is on the $j'th$ peg.

The disk with index 1 refers to the smallest disk, and the disk with index n refers to the largest disk.

Perfect state A vertex in the Hanoi graph represents a perfect state if, for some value j, all $a_i = j$. This means all disks are located on the j-th pole. In the Hanoi graph H_n^k, there are k perfect states.

Shortest Path through Hanoi Graph Solving the Tower of Hanoi puzzle in the fewest moves possible corresponds to finding the shortest path between two vertices in the Hanoi graph.

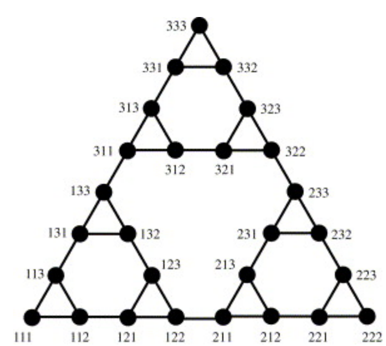

Fig. 2. The state graph for (R) H_3^3

Figure 2 illustrates the shortest path in H_3^3 connecting the configurations 111 and 333. The length of this path is consistent with the previously derived result for the optimal number of moves:

$$T(3) = 2^3 - 1 = 7.$$

3 Sierpiński Triangle, Pascal's Triangle, and beyond

Fig. 3. Sierpiński Triangle

3.1 Analogy with Sierpiński Triangle

The Tower of Hanoi puzzle exhibits a surprising connection to the Sierpiński triangle. Ian Stewart first identified that the graph of the Tower of Hanoi puzzle, especially when visualized for n disks, mirrors the recursive structure of the Sierpiński triangle.

Recursive decomposition of the Hanoi graph For n disks, the puzzle graph can be recursively decomposed into smaller instances of the graph for $n-1$ disks. This recursive decomposition reflects how the Sierpiński triangle is constructed by recursively removing smaller triangles from a larger one.

The Hanoi graph for n disks consists of three copies of the graph for $n-1$ disks, each connected by edges representing valid moves between configurations. As n increases, this recursive structure grows like the recursive construction of the Sierpiński triangle, where each recursive step scales down the size of the problem (or the length of the side of each triangle) by a factor of two.

Sierpiński triangle is a fractal pattern (refer to Figure 3) that exhibits *self-similarity*. It is constructed by recursively subdividing an equilateral triangle into four smaller equilateral triangles and removing the central one. This process is repeated infinitely for each of the remaining triangles. The resulting pattern consists of nested triangles, forming a visually striking geometric figure.

The Sierpiński Triangle can also be formed in a manner analogous to the Cantor set. Starting with a solid equilateral triangle, at each stage, we divide each solid triangular portion into four smaller triangles and remove the central one.

Furthermore, taking the bottom-left section (one-third of the total) and doubling its length in each direction gives us three copies of the bottom-left section.

Using this property, the fractal dimension d of the Sierpiński Triangle can be calculated as:

$$2^d = 3 \quad \text{or} \quad d = \frac{\log 3}{\log 2} \approx 1.585$$

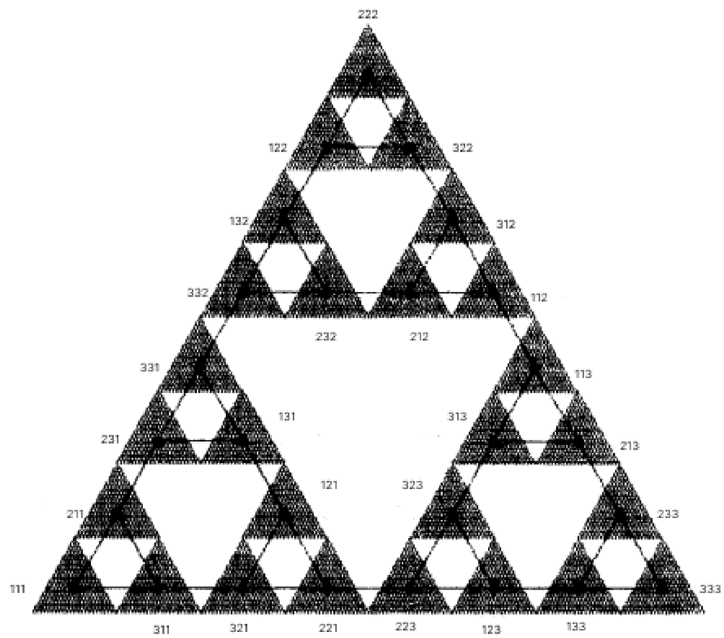

Fig. 4. Sierpiński triangle isomorphic to (L) H_3^3

The isomorphism between Sierpiński triangle with granularity n and a 3-peg Hanoi graph with n disks has been shown in the study by Hinz and Schief [6]. When n grows without bound, H_n^3 comes to resemble the famous Sierpiński triangle, demonstrating its fractal structure.

3.2 Connection to Pascal's Triangle and Binary Representation

The connection between the Tower of Hanoi and the Sierpiński triangle deepens through the use of Pascal's triangle and binary representation. This relationship can be explained through Lucas' Theorem, which helps determine the parity (odd or even) of binomial coefficients $C(n, m)$.

Pascal's Triangle is a triangular arrangement of binomial coefficients [1], where each entry $C(n, m)$ represents the number of ways to choose m elements from a set of n elements. It is constructed as follows:

– The first row contains a single 1.
– Each entry is the sum of the two directly above it.

According to **Lucas' Theorem**, the binomial coefficient $C(n,m)$ is odd if and only if, for each binary digit of n and m, there are no overlapping 1s in the same positions. This parity condition directly influences both the structure of the Tower of Hanoi puzzle and the construction of the Sierpiński triangle.

This binary relationship mirrors the construction of the Sierpiński triangle, where the binary representation of row and column indices determines the parity of entries in Pascal's triangle. The absence of overlapping 1s in the binary representation plays a crucial role in shaping the resulting structure.

Thus, it turns out that the graph H_n^3 is closely connected to Pascal's triangle. For instance, the graph of the odd entries in the first eight rows of Pascal's triangle is isomorphic to H_3^3 (See Figure 5). By considering additional rows of Pascal's triangle, we observe that the Hanoi graphs H_n^3 are each isomorphic to sub graphs of Pascal's triangle.

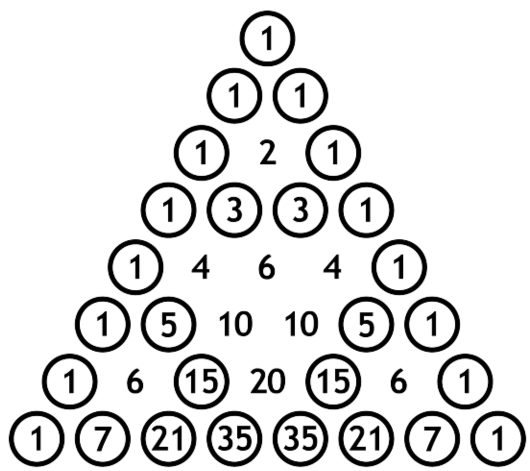

Fig. 5. TOH(3,3) Pascal's triangle with odd binomial coefficients encircled

Mathematical Foundation The binomial coefficient $C(n,m)$, which calculates the number of ways to select m elements from a set of n elements, is given by the formula:

$$C(n,m) = \frac{n!}{m!(n-m)!}$$

Lucas' Theorem provides a criterion for determining whether $C(n, m)$ is odd based on the binary representations of n and m. Specifically, Lucas' Theorem states that:

$$C(n, m) \text{ is odd } \iff A_m \subseteq A_n$$

where A_n and A_m represent the sets of positions of 1s in the binary representations of n and m, respectively.

If there is no overlap of 1s between the binary representations of n and m, then $C(n, m)$ is odd, corresponding to a valid move in the Tower of Hanoi puzzle and contributing to the construction of the Sierpiński triangle.

3.3 The Special Constant 466/885

Finding the shortest path from a *perfect* initial state to a *perfect* final state is well-known. However, it becomes more complex when dealing with arbitrary initial and final states that are not *perfect*. Here, the problem reduces to identifying the shortest path in the Hanoi graph.

Behavior of the largest separating disk A key challenge is determining how to optimally move the largest disk that "separates" the initial and terminal states—i.e., the largest disk that is not positioned on the same peg in both the initial and final configurations. As observed in prior work, including [7], the question reduces to deciding whether it needs to be moved once or twice, which influences the length of optimal path.

Relationship to the Sierpiński Gasket As discussed in earlier subsections, as the number of disks, n, approaches infinity, the Tower of Hanoi graph becomes isomorphic to the Sierpiński Gasket. This isomorphism reveals a constant which represents the asymptotic average length (with respect to $2^n - 1$) of the shortest path between two random states in the Hanoi graph. For convenience, this constant will henceforth be referred to as the *Hanoi* constant.

The value of the Hanoi constant is found, as in [6], to be equal to $\frac{466}{885} \approx 0.52655$.

This value mirrors the average distance (with respect to the diameter) between two randomly chosen points on the Sierpiński Gasket, highlighting a deep geometric connection between the puzzle and fractal structures.

Implications of the Hanoi constant The Hanoi constant implies that, on average, moving from one random configuration to another requires only about half as much effort as solving the hardest possible path of length $2^n - 1$ in the classic Tower of Hanoi problem.

4 Extension to 4-Peg Tower of Hanoi

The 4-peg Tower of Hanoi problem builds upon the classical 3-peg version. When an additional peg is introduced, the problem becomes significantly more complex. While the 3-peg version has a well-defined recursive solution, the optimal strategy for the 4-peg version remained uncertain for a long time. The challenge lies in leveraging the additional flexibility the fourth peg provides without excessively increasing the solution's complexity.

4.1 Graph Representations

The graph H_n^4 expands the idea of the connection of Hanoi graphs with Sierpiński triangle and Pascal triangle (as discussed in Section 3), but introduces additional complexity. Unlike the 3-peg case, the 4-peg Hanoi graph does not directly correspond to a simple fractal like the Sierpiński triangle. Attempts to relate it to higher-dimensional analogues, such as Pascal's pyramid, have been less successful, as the increased number of connections and possible moves disrupt the direct self-similar patterns observed in the 3-peg problem.

Hanoi Graph $\boldsymbol{H_1^4}$ The graph H_1^4 (see Figure 8) represents the state space of the 4-peg Tower of Hanoi when there is only one disk ($n = 1$). In this scenario:

- There are 4 vertices, each representing a configuration where the single disk is placed on one of the 4 pegs.
- Since the disk can move freely between any pegs, the graph H_1^4 is equivalent to the complete graph K_4.
- The graph has 6 edges, corresponding to all possible moves between the four pegs:
$$\text{Edges} = \binom{4}{2} = 6.$$
- H_1^4 is planar and can be visualized easily as a triangle with a central node connected to all other nodes.

This simple structure serves as the base case for constructing more complex Hanoi graphs as the number of disks increases.

Hanoi Graph $\boldsymbol{H_2^4}$ The graph H_2^4 (see Figure 6) extends this idea to two disks, significantly increasing the complexity:

- There are 16 vertices, each representing a distinct configuration of the two disks across the 4 pegs. A configuration can be described by the pair (a, b), where a is the position of the larger disk and b is the position of the smaller disk.
- The graph structure reflects the legal moves between configurations:
- The construction of H_2^4 involves:

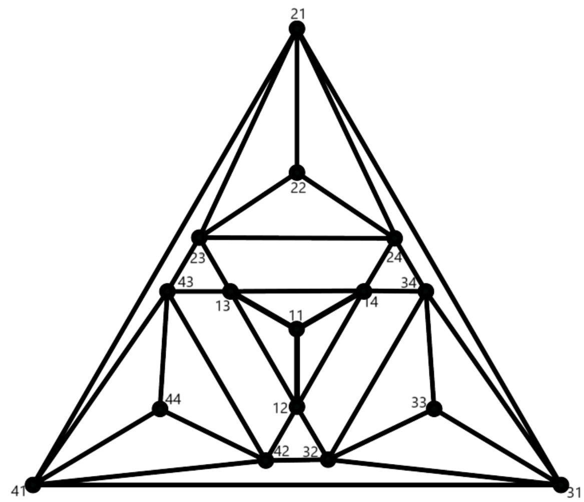

Fig. 6. Planar graph of (R) H_2^4

- Subgraphs for each fixed position of the larger disk. There are four subgraphs, each resembling the smaller H_1^4 graph for the single smaller disk.
- Bridges that connect these subgraphs allow transitions when the larger disk moves. By a combinatorial argument [9], the total number of bridges in H_k^n is given by

$$\binom{k}{2}(k-2)^{n-1}.$$

– The total number of edges in H_2^4 can be calculated as:

$$\text{Edges} = (4 \times 6) + (12) = 36.$$

Here:
- 4×6 accounts for the edges within the sub graphs.
- 12 accounts for the bridges between sub graphs.

These initial cases of the 4-peg puzzle illustrate the growth in complexity as the number of disks increases. While H_1^4 is simple and fully connected, H_2^4 introduces substructures and connections that set the stage for more intricate configurations in higher cases.

4.2 Multiple Shortest Paths

In the 4-peg puzzle, the introduction of an additional peg creates more flexibility in the solution space, leading to multiple shortest paths between certain configurations. Unlike the 3-peg version, where the shortest path is *unique* and follows a predictable recursive pattern, the extra peg provides additional options for intermediate moves. This results in several alternative routes to reach the goal state in the same number of moves.

For example, with two disks, there are multiple valid shortest paths, such as:

$$11 \to 12 \to 42 \to 44$$

and

$$11 \to 13 \to 43 \to 44$$

Both paths take the same number of moves, but the intermediate steps differ, demonstrating the non-uniqueness of the solution in the 4-peg scenario. To find a solution to the 4-peg Tower of Hanoi puzzle, we attempted a naive approach, which is recursive in nature, as detailed in the following section.

4.3 Recursive Solution (Naive Approach)

Let the four pegs be labeled as *Src*, *Aux1*, *Aux2*, and *Dst*. Initially, all disks are stacked on *Src*, and the goal is to move all disks to *Dst*. For any number n of disks, the solution involves these key steps:

1. Move the top $n-2$ disks from *Src* to *Aux1* (using *Aux2* and *Dst* as intermediaries).
2. Move the second largest disk from *Src* to *Aux2*.
3. Move the largest disk from *Src* to *Dst*.
4. Move the second largest disk from *Aux2* to *Dst*.
5. Move the $n-2$ disks from *Aux1* to *Dst* (using *Src* and *Aux2* as intermediaries).

```
Solve(N, Src, Aux1, Aux2, Dst)

if N = 0 then
   exit
else if N = 1 then
   Move from Src to Dst
else
   Solve(N - 2, Src, Dst, Aux2, Aux1)
   Move from Src to Aux2
   Move from Src to Dst
   Move from Aux2 to Dst
   Solve(N - 2, Aux1, Aux2, Src, Dst)
end if
```

Recurrence Relation Let $T(n)$ represent the minimum time required to transfer n disks from *Src* to *Dst*. Based on the algorithm outlined above, the recurrence relation for this problem is:

$$T(1) = O(1) \quad \text{(base case)}$$

$$T(n) = 2T(n-2) + O(1) \quad \text{(recursive case)}$$

This relation provides an upper bound on the minimum time needed to solve the Tower of Hanoi problem with 4 pegs and n disks. Solving the recurrence relation yields the expression:

$$T(n) = O(2^{\frac{n}{2}}),$$

and the minimum time required grows exponentially with the number of disks. However, the rate of growth is slower than that of the 3-peg Tower of Hanoi approach.

5 Beyond the current research (Bonus)

5.1 The extensions of Tower of Hanoi in multiple dimensions

As seen in the previous sections, there exists an elegant way to show every possible move order of the Tower of Hanoi game in a [5] planar 2-Dimensional graph. Some involving modulus [1], and som using methods of binary counting. To extend this idea into the domain of multiple dimensions, there are some base ideas from the Sierpiński's triangle and tetrahedral we would be using. The main property of which being that, similar to a space filling curve, the pattern is infinitely repeatable. In our case, this means that the H_n^k must be able to be constructed from H_{n-1}^k. This repeatability is a major part of the figure being a cantor set [6].

The next part which needs to be concretized is an intuition of symmetry that exists in the graphs. This would mean that we can make a group [4] of symmetries in our graph, let that group be labeled as D_3^2. This symbol would indicate it is the group of symmetries of a regular 3-sided, 2-dimensional fractal variation of Sierpiński's triangle. Let the elements of this group be $\{\rho_0, \rho_1, \rho_2, \mu_1, \mu_2, \mu_3\}$. Where ρ_i is used to indicate rotations and μ_i is for mirror images taken along angle bisectors of the triangle. We can therefore define the Group table for D_3^2 as the following.

The proof of D_3^2 being a group is as follows.

Proof. **1. Proof of Associativity**

Through the group table, we observe that:

$$\forall a, b, c \in D_3^2, \quad (a \circ b) \circ c = a \circ (b \circ c).$$

This follows from the fact that each symmetric action on the figure results in a symmetric figure.

2. Proof of Existence of Identity

$\circ$	ρ_0	ρ_1	ρ_2	μ_1	μ_2	μ_3
ρ_0	ρ_0	ρ_1	ρ_2	μ_1	μ_2	μ_3
ρ_1	ρ_1	ρ_2	ρ_0	μ_3	μ_1	μ_2
ρ_2	ρ_2	ρ_0	ρ_1	μ_2	μ_3	μ_1
μ_1	μ_1	μ_2	μ_3	ρ_0	ρ_1	ρ_2
μ_2	μ_2	μ_3	μ_1	ρ_2	ρ_0	ρ_1
μ_3	μ_3	μ_1	μ_2	ρ_1	ρ_2	ρ_0

Table 1. Group table for D_3^2

This property states that:

$$\exists! e \in D_3^2, \quad \forall a \in D_3^2, \quad a \circ e = e \circ a = a.$$

From the group table, it can be observed that ρ_0 acts as the identity element. Here, ρ_0 can be interpreted as a 360-degree rotation.

3. Proof of Existence of Inverse
This property states that:

$$\forall a \in D_3^2, \quad \exists! a' \in D_3^2, \quad a \circ a' = a' \circ a = e \quad \text{(where } e \text{ is the identity).}$$

For mirror images, the inverse is the element itself (applying the mirror image operation twice results in the original figure). For rotations, if an element represents a rotation by x degrees, the inverse is a rotation by $(360 - x)$ degrees. [2]

Thus, D_3^2 satisfies all the group axioms and is a group. □

We thus have a notion of similarity between the structure of a graph we produce and a finite 'granularity' Sierpiński's triangle. We shall build our intuition of the graphs in higher dimensions upon this proposition,

Proposition 1. *If a graph of H_n^k is isomorphic to a Sierpiński gasket, there exists an arrangement of the graph with the following properties:*

1. *It can be built from the graphs of H_{n-1}^k for all $n > 1$. That is, there is existence of graphical induction.*
2. *It has a group of symmetries isomorphic to a group D_q^p for some p and q.*

With this proposition we can build more concrete similarities between Hanoi Graphs and Sierpiński's gasket. We can test this proposition for our previous knowledge of graphs of H_n^3. For example, we consider the graph of H_4^3 as given in Figure 7.

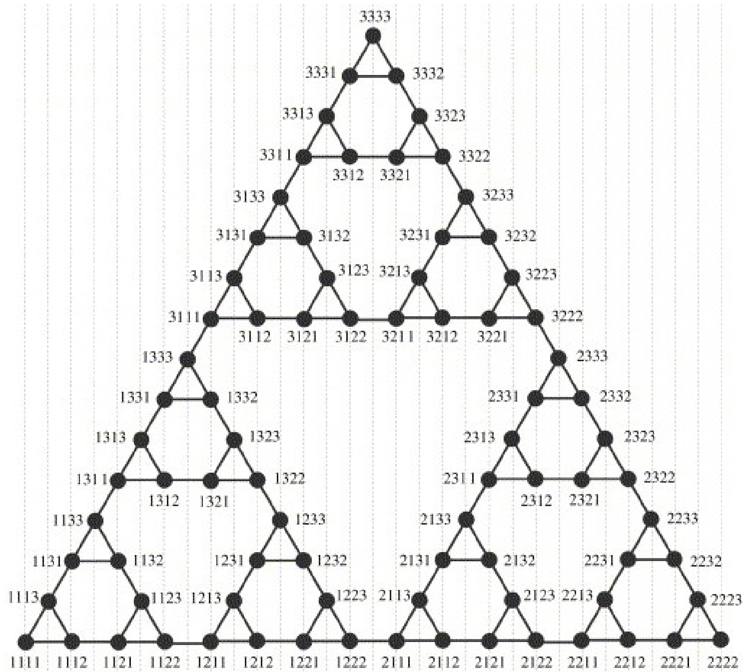

Fig. 7. The state graph for (R) H_4^3

Here the states are given as (1 indexed strings) such that if the $i'th$ number of the state (from the right) is j, that would mean that in the current state of the puzzle, the $i'th$ disk is on the $j'th$ peg. This notation is valid in Figure 7 and 2. The notation in the state diagram for the remainder of the paper takes i from the left. It is clear to see and verify the proposition. For verification of the first part of the proposition, the graph of H_3^3 is given in Figure 2.

We can see that the H_3^4 graph is a replication of 3 such H_3^3 graphs connected via bridges. The verification of the second proposition requires more rigorous proving. We can consider the following isomorphism with D_3^2.

$\rho_0 \rightarrow$ Rotation of triangle by 360 degrees

$\rho_1 \rightarrow$ Rotation of triangle by 120 degrees

$\rho_2 \rightarrow$ Rotation of triangle by 240 degrees

$\mu_1 \rightarrow$ Mirror image about the bisector of the triangle through state 3333

$\mu_2 \rightarrow$ Mirror image about the bisector of the triangle through state 1111

$\mu_3 \rightarrow$ Mirror image about the bisector of the triangle through state 2222

Thus our proposition 1 is in accordance with examples of the graphs in lower dimensions and previous research.

Now the process of extending this to higher dimensions starts. This would require us to start with taking a look at the immediate next graph of Tower of Hanoi as the graph of H_n^3 is well documented and studied. This would mean trying to map out the graph of H_n^4 and begin the foundation of our studies.

[3] We can begin by analyzing the smallest concrete case of H_n^4 we have, namely H_1^4 and H_2^4.

H_1^4 is a seemingly trivial case we see for the puzzle where every state we have is a final state. Thus our graph, if made in a planar fashion would look like figure 8. This however gives us a good baseline for our inductive hypothesis part for the proposition, implying that all following H_n^4 graphs would contain H_1^4 as the building blocks.

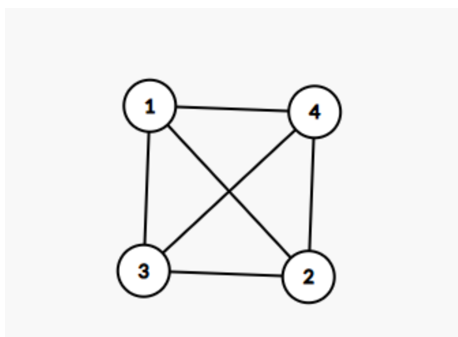

Fig. 8. The state graph for (R) H_1^4

The graph for H_2^4 however provides a sudden jump in complexity as expected and shows the difficulty of having path finding algorithms as the number of discs increase with more pegs. The graph shown in Figure 9, gives a basic intuition of there existing a property of graphical induction here. We can see that the points $(11, 21, 31, 41)$, $(12, 22, 32, 42)$, $(13, 23, 33, 43)$ and $(14, 24, 34, 44)$ form cliques which can be thought of as H_1^4 graphs. These H_1^4 graphs are connected to one another with bridges.

For analysis of H_2^4 in the sense we are intending, we would have to ignore the notion of it being a planar graph and take the figure into a higher dimension, in this case, the 3rd dimension. Doing this with alignment of the graph being a traditional Sierpiński Tetrahedron gives us the following approximation of shape to the graph as an extension of the graph in figure 10.

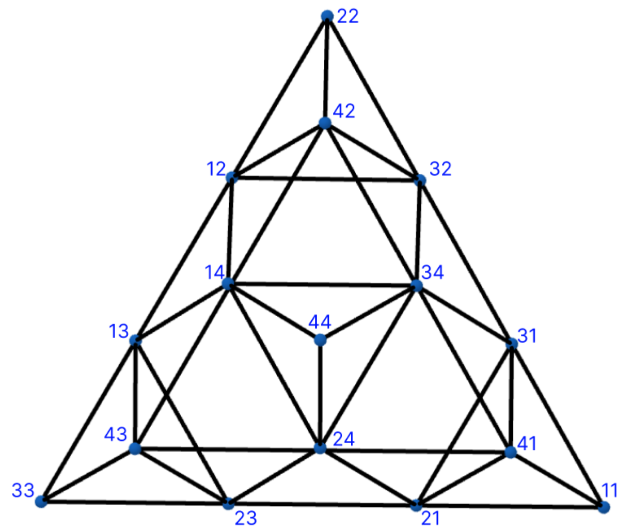

Fig. 9. The state graph for (L) H_2^4

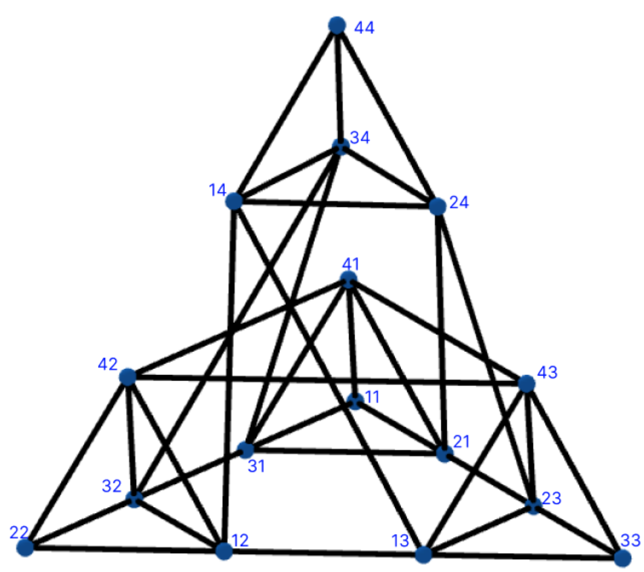

Fig. 10. The state graph for (L) H_2^4 in 3 Dimensions sticking to a Sierpiński gasket shape

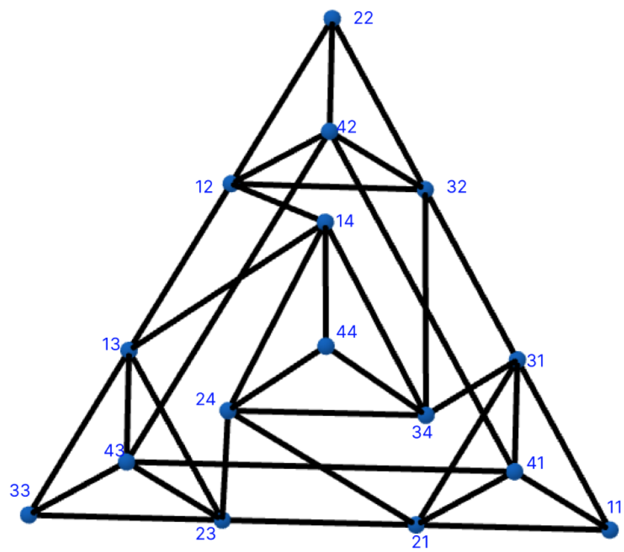

Fig. 11. The arial view of the graph shown in Figure 10

The graph made in Figure 10 preserves the inductive property but it lacks behind in the second property as it isn't able to produce the symmetries that exist with regular triangular pyramids. Thus we cannot safely assume that this a correct configuration of the graph for our analysis purposes. Thus to try and modify our graph we would have to stray from the traditional Sierpiński tetrahedral. We were allowed a bit of leeway before due to the existence of the bridges but now we need to take it a step further and apply rotations on parts of the graph to gain more symmetries without disturbing its inductiveness.

Figure 12 is the option we get by applying a rotation of 60 degrees on the top-most subsection of graph. There are a few desirable properties about this configuration that we will expand upon.

1. The Top view of the graph produces the 2 dimensional version of the graph we see in figure 9. This property can be looked upon as an analogous of 'projections'.
2. The inductive nature is apparent as we maintain symmetric cliques of H_1^4's.
3. The graph now behaves like a regular tetrahedron implying an isomorphism existing between the group of symmetries of a regular Sierpiński Tetrahedral, in our notation this would be D_4^3 as we are talking about the variation of regular Sierpiński's Gasket with 4 vertices in 3 dimensions. [8]

Using this we can infer that the rotation of a part of the graph has preserved the main identifying properties of the graph while adding a few helpful tweaks

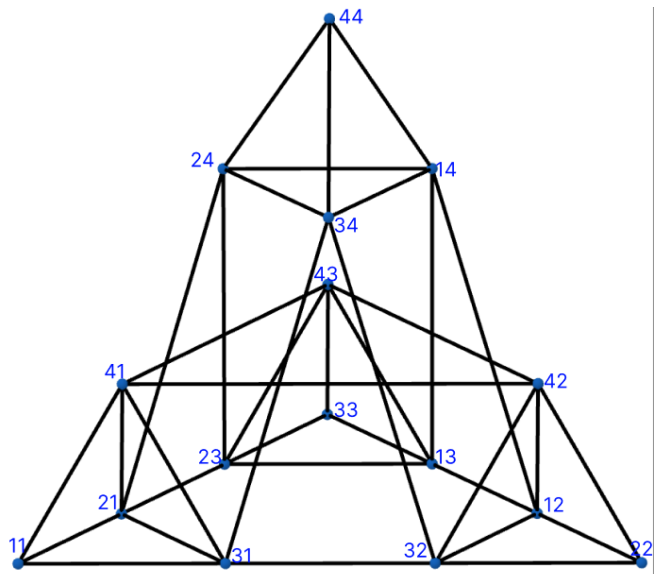

Fig. 12. The state graph for (L) H_2^4 in 3 Dimensions with modifications

to our system.

Here we identify another extremely interesting fact however, the first point on the list is something we could not see at the scale of only looking at the puzzle from the second dimension but is a luxury we can afford with the third dimension. This shows that the level set of the discrete graph (or the projection/shadow) of the graph on a lower dimension surface will not only be an accurate representation of the graph, but it would also follow the requirements of proposition 1 but in a slightly modified manner. We will state that formally now.

Proposition 2. *If a graph of H_n^k is isomorphic to a t dimensional Sierpiński fractal and $k > 3$, the graph of H_n^k projected upon a surface of dimension $k-2$ will have the following properties*

1. *It can be built from the graphs of H_{n-1}^k projected upon a surface of dimension $k-2$ for all $n > 1$. That is, there is existence of graphical induction.*
2. *It has a group of symmetries isomorphic to a group D_q^p for some p and q.*

The main idea of proposition 2 is to show that the task of going from one dimension to the next is one that can employ the game of Tower of Hanoi to solve.

Another proposition we have discusses the structure of the graphs itself. Let us begin with observations we can see and try to concretize them with proofs.

Proposition 3. *The base case of every H_n^k graph is a k-clique with each state in the clique having the $i'th$ terms equal $\forall i > 1$.*

Corollary 1. *Every state in the diagram belongs to a single k-clique.*

Proof. As the smallest disc is on the top of the peg it is on in any valid configuration of the puzzle, we can say that the smallest peg being moved to any of the other $k-1$ pegs will be a valid move and will always available. Thus there are sets of k states which will always exist. We can thus say that the building block of the graph of H_n^k are k-cliques. □

With all of the theory given here, using the knowledge of k-cliques existing as the basic decomposition parts of the entire graph, we can reach a concluding theory for this section that has been inductively proved from all dimensions, including higher ones that aren't covered in literature.

Theorem 1. *An n dimensional Sierpiński fractal can be considered isomorphic to a game of Tower of Hanoi with $n+1$ pegs.*

5.2 Optimized algorithm for solving 4-Peg Tower of Hanoi with improved time complexity

The minimum number of moves required to solve the 4-peg Tower of Hanoi problem with n disks is significantly lower compared to the 3-peg version. However, the naive approach is simplistic, failing to fully leverage the additional auxiliary peg. To address this limitation, a refined method has been developed and is presented in detail below.

The objective of this modified approach is to optimize the algorithm's time complexity without increasing the total number of moves required. The approach integrates the classical 3-peg Tower of Hanoi algorithm with a tailored strategy for the 4-peg version.

This section outlines the proposed algorithm, followed by a comprehensive analysis of its time complexity. Finally, a comparative analysis between the novel approach and the naive approach is presented in a graphical form.

Algorithm Let the four pegs be labeled as *Src*, *Aux1*, *Aux2*, and *Dst*. Initially, all disks are stacked on *Src*, and the goal is to move all disks to *Dst*. For any number n of disks, the solution involves these key steps:

Here, $j = \frac{n}{1.5}$, $k = n - j$, $j_1 = \frac{j}{2}$, $j_2 = j - j_1$

1. Move the first j_1 disks from *Src* to *Aux2* using *Aux1*.
2. Move the next j_2 disks from *Src* to *Aux1*, using *Dst*.
3. Move the j_1 disks from *Aux2* to *Aux1* using *Dst*.
4. Move the remaining k disks directly from *Src* to *Dst* using *Aux2*.
5. Move the j_1 disks back from *Aux1* to *Src* using *Aux2*.

6. Move the j_2 disks from *Aux1* to *Dst* using *Aux2*.
7. Finally, move the j_1 disks from *Src* to *Dst* using *Aux2*.

```
TOH_3_PEG(N, Src, Aux, Dst)

if N = 0 then
   exit
else
   Move from Src to Dst
   Solve(N − 1, Aux, Src, Dst)
end if
```

```
TOH_4_PEG(N, Src, Aux1, Aux2, Dst)

if N = 0 then
   exit
else
   j = N/1.5
   k = N − j
   j1 = j/2
   j2 = j − j1
   TOH_3_PEG(j1, Src, Aux1, Aux2)
   TOH_3_PEG(j2, Src, Dst, Aux1)
   TOH_3_PEG(j1, Aux2, Dst, Aux1)
   TOH_3_PEG(k, Src, Aux2, Dst)
   TOH_3_PEG(j1, Aux1, Aux2, Src)
   TOH_3_PEG(j2, Aux1, Aux2, Dst)
   TOH_3_PEG(j1, Src, Aux2, Dst)
end if
```

This algorithm efficiently uses the fourth peg to move the first set of disks. However, due to structural limitations, it switches to using only three pegs for the remaining disks. Once those are moved, it again uses four pegs to transfer the final disks to their destination.

Time Complexity Analysis Let $T(n)$ represent the minimum time required to transfer n disks from *Src* to *Dst* with 4 pegs. Let $R(n)$ represent the minimum time required to transfer n disks from *Src* to *Dst* with 3 pegs. From earlier analysis, it is known that $R(n)$ is $O(2^n)$.

Based on the algorithm outlined above, the recurrence relation for this problem is:

$$T(1) = O(1) \quad \text{(base case)}$$

$$T(n) = 6R\left(\frac{n}{3}\right) + R\left(\frac{n}{3}\right) \quad \text{(recursive case)}$$

Solving the recurrence relation yields the expression:

$$T(n) = O\left(2^{\frac{n}{3}}\right)$$

The results demonstrate that the time required increases exponentially with the number of disks, though the rate of growth is slower compared to the previous naive approach.

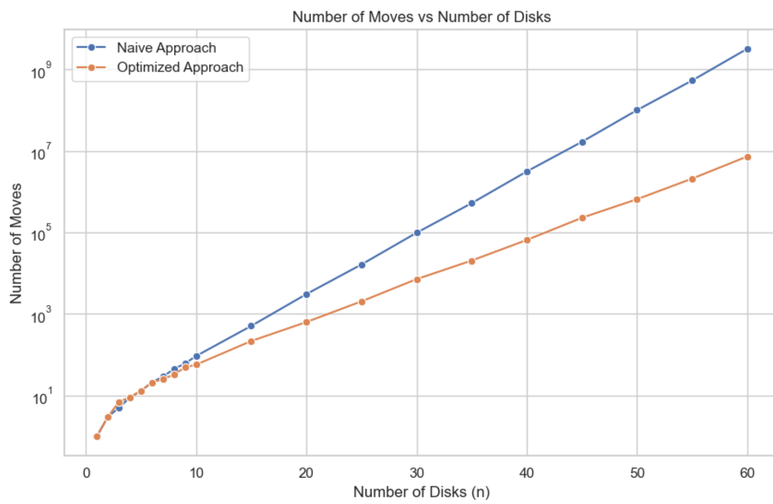

Fig. 13. Graphical Comparison of Naive and Optimized Approaches

Comparative Analysis of the Naive and Optimized Approaches Figure 13 shows that the optimized approach is much more efficient than the naive approach, especially as the number of disks increases. For smaller values of n, both perform similarly, but as n gets larger, the optimized approach requires far fewer moves. For example, when n=60, the naive approach needs over 3.2 billion moves, while the optimized approach reduces this to around 7.3 million. This demonstrates that the optimized approach is more practical for complex problems.

References

[1] Tom Bannink and Harry Buhrman. *Quantum Pascal's Triangle and Sierpinski's carpet*. 2017. arXiv: `1708.07429 [quant-ph]`. URL: `https://arxiv.org/abs/1708.07429`.

[2] Mary L. Boas. *Mathematical Methods in the Physical Sciences*. Wiley, 1983. ISBN: 978-0471198260.

[3] Stephanie Egler. "Graphs, Random Walks, and the Tower of Hanoi". In: *Rose-Hulman Undergraduate Mathematics Journal* 20.1 (2019), Article 6. URL: `https://scholar.rose-hulman.edu/rhumj/vol20/iss1/6`.

[4] JB. Fraleigh. *A first course in Abstract Algebra*. Pearson, 2014. ISBN: 9789332519039.

[5] Andreas M. Hinz. "Metric properties of the Tower of Hanoi graphs and Stern's diatomic sequence". In: *European Journal of Combinatorics*. 2005.

[6] Andreas M. Hinz and Andreas Schief. "The Average Distance on the Sierpiński Gasket". In: *Probability Theory and Related Fields* 87.1 (Mar. 1990), pp. 129–138. ISSN: 1432-2064. DOI: `10.1007/BF01217750`. URL: `https://doi.org/10.1007/BF01217750`.

[7] Dan Romik. *Shortest paths in the Tower of Hanoi graph and finite automata*. 2003. arXiv: `math/0310109 [math.CO]`. URL: `https://arxiv.org/abs/math/0310109`.

[8] Orlin Stoytchev. "Topology of SO(3) for Kids". In: (2023). URL: `https://arxiv.org/pdf/2310.19665`.

[9] Andrew Zhang. "A Stratification of the Hanoi Graph for 4 Pegs". In: 2009. URL: `https://api.semanticscholar.org/CorpusID:19351380`.